



## Laboratório 4 - Grafos

Semana de 7 a 11 de Novembro de 2016

Duração: 2 horas

Neste laboratório abordam-se os tópicos da representação e manipulação de grafos. Na 1ª parte recorre-se à representação em matriz de adjacências, sendo solicitado o cálculo de algumas propriedades de um grafo e a produção de um ficheiro de saída com todos os ramos do grafo. Na 2ª parte pretende-se usar a representação em lista de adjacências.

### 1ª Parte

Considere um tipo de ficheiro de dados em formato texto, com a extensão `adj`,

`<nome_ficheiro_matriz_adjacencias>.adj`

em que cada ficheiro contém a descrição de um grafo não direccionado, ponderado, usando a representação em matriz de adjacências. A primeira linha tem um inteiro,  $n$ , que indica o número de nós (ou vértices) do grafo. A matriz de adjacências (simétrica) é dada pelas  $n$  linhas seguintes, cada uma com  $n$  colunas de inteiros não negativos, em que um valor positivo na posição  $(i, j)$  indica o peso da aresta entre o nó  $i$  e o nó  $j$ , sendo que o valor 0 na posição  $(i, j)$  indica a ausência de aresta entre o nó  $i$  e o nó  $j$ . Os nós são numerados de 0 a  $n-1$ .

Escreva um programa, designado `lab4a`, invocado pelo comando

`lab4a <nome_ficheiro_matriz_adjacencias>.adj`

que lê o ficheiro `<nome_ficheiro_matriz_adjacencias>.adj` e implementa as seguintes tarefas:

1. Constrói, a partir dos dados lidos, uma estrutura de dados que representa o grafo, usando a representação em matriz de adjacências.
2. Imprime, no écran, a densidade do grafo.
3. Imprime, no écran, o grau máximo do grafo.
4. Escreve no ficheiro de saída em formato texto, com a extensão `ram`,

`<nome_ficheiro_matriz_adjacencias>.ram`

todos os ramos do grafo. A primeira linha do ficheiro deve indicar o número de nós,  $n$ , a segunda o número de ramos,  $r$ , e as  $r$  linhas seguintes devem conter os nós de cada ramo seguidos do peso do ramo, sendo descrito um ramo por linha.

Na Fig. 1 apresenta-se um exemplo de um ficheiro de dados, cuja densidade é 1.666... e grau máximo é 3. Na figura 2 apresenta-se o ficheiro de saída correspondente.

```

6
0 2 1 3 0 0
2 0 0 0 0 0
1 0 0 4 5 0
3 0 4 0 0 0
0 0 5 0 0 0
0 0 0 0 0 0

```

Figura 1: Exemplo de um ficheiro de dados: grafo6.adj

```

6
5
0 1 2
0 2 1
0 3 3
2 3 4
2 4 5

```

Figura 2: Exemplo de um ficheiro de saída: grafo6.ram

## 2ª Parte

Na 2ª parte pretende-se fazer uso da representação de grafos em lista de adjacências. Escreva um programa, designado `lab4b`, invocado com o comando

```
lab4b <nome_ficheiro_lista_amos>.ram
```

que lê um ficheiro do tipo `.ram`, como exemplificado na Fig. 2, que contém as arestas de um grafo não direccionado, ponderado, e que implementa as seguintes tarefas:

1. Constrói, a partir dos dados lidos, uma estrutura de dados que representa o grafo, usando a representação em lista de adjacências.
2. Escreve um ficheiro de saída, com a extensão `ladj`,

```
<nome_ficheiro_lista_adjacencias>.ladj
```

com uma representação gráfica da lista de adjacências, como exemplificado na Fig. 3. A primeira linha deve indicar o número de nós do grafo, descrevendo as linhas seguintes a lista de vértices adjacentes de cada vértice. Quando um vértice é isolado, a linha contém -1. Para cada vértice adjacente, deve ser indicado primeiro o vértice, seguido do peso da aresta, separado por `:`. As linhas são terminadas com -1.

Nos ficheiros `LinkedList.h` e `LinkedList.c` encontram-se definidos os protótipos para manipulação de uma lista simples de `Items` e as suas implementações, respectivamente, que deverá usar no seu programa. A implementação está apenas parcialmente feita e deverá ser completada no laboratório. O tipo `Item` é definido no ficheiro `defs.h`.

**Escreva um ficheiro `makefile` que permita a compilação do programa.**

```

6
1:2 2:1 3:3 -1
0:2 -1
0:1 3:4 4:5 -1
0:3 2:4 -1
2:5 -1
-1

```

Figura 3: Exemplo de um ficheiro de saída: grafo6.ladj